Configuring Network Security

This chapter contains network security information unique to the Cisco 7600 series routers, which supplements the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.1, at this URL:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/index.htm
- *Cisco IOS Security Command Reference*, Release 12.1, at this URL:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_r/index.htm

This chapter consists of these sections:

Note With Releases 12.1(11b)E and later releases, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

ACL Configuration Guidelines

The following guidelines apply to ACL configurations:

- Each type of ACL (IP, IPX, and MAC) filters only traffic of the corresponding type. A MAC ACL never matches IP or IPX traffic.
- By default, the MSFC sends Internet Control Message Protocol (ICMP) unreachable messages when a packet is denied by an access group.

With the **ip unreachables** command enabled (which is the default), a Supervisor Engine 2 drops most of the denied packets in hardware and sends only a small number of packets to the MSFC2 to be dropped (10 packets per second, maximum) , which generates ICMP-unreachable messages.

With the **ip unreachables** command enabled, a Supervisor Engine 1 sends all the denied packets to the MSFC to be dropped, which generates ICMP-unreachable messages. With a Supervisor Engine 1, to drop access list-denied packets in hardware, you must disable ICMP-unreachable messages using the **no ip unreachables** interface configuration command.

To eliminate the load imposed on the MSFC CPU by the task of dropping denied packets and generating ICMP-unreachable messages, do the following:

- With Supervisor Engine 1, enter the **no ip unreachables** interface configuration command.
- With Supervisor Engine 2, enter the **no ip unreachables** and the **no ip redirects** interface configuration commands. (CSCdr33918)

- ICMP unreachable messages are not sent if a packet is denied by a VACL.

Hardware and Software ACL Support

Access control lists (ACLs) can be processed in hardware by the Policy Feature Card (PFC or PFC2), the Distributed Forwarding Card (DFC), or in software by the Multilayer Switch Feature Card (MSFC or MSFC2). The following behavior describes software and hardware handling of ACLs:

- ACL flows that match a “deny” statement in standard and extended ACLs (input and output) are dropped in hardware if “ip unreachables” is disabled.
- ACL flows that match a “permit” statement in standard and extended ACLs (input and output) are processed in hardware.
- VLAN ACL (VACL) flows are processed in hardware. If a field specified in a VACL is not supported by hardware processing that field is ignored (for example, the **log** keyword in an ACL) or the whole configuration is rejected (for example, a VACL containing unsupported IPX ACL parameters).
- VACL logging is processed in software.
- Dynamic ACL flows are processed in the hardware; however, idle timeout is processed in software.
- IP accounting for an ACL access violation on a given port is supported by forwarding all denied packets for that port to the MSFC for software processing without impacting other flows.
- Extended name-based MAC address ACLs are supported in hardware.
- The following ACL types are processed in software:
  - Standard XNS access list
  - Extended XNS access list
  - DECnet access list
  - Internetwork Packet Exchange (IPX) access lists
  - Extended MAC address access list
  - Protocol type-code access list

Note IP packets with a header length of less than five will not be access controlled.

- Flows that require logging are processed in software without impacting nonlogged flow processing in hardware.
- The forwarding rate for software-processed flows is substantially less than for hardware-processed flows.
- When you enter the **show ip access-list** command, the match count displayed does not include packets processed in hardware.

Guidelines and Restrictions for Using Layer 4 Operators in ACLs

These sections describe guidelines and restrictions when configuring ACLs that include Layer 4 port operations:

- Determining Layer 4 Operation Usage, page 23-3
- Determining Logical Operation Unit Usage, page 23-4

Determining Layer 4 Operation Usage

You can specify these types of operations:

- gt (greater than)
- lt (less than)
- neq (not equal)
- eq (equal)
- range (inclusive range)

We recommend that you do not specify more than *nine different* operations on the same ACL. If you exceed this number, each new operation might cause the affected ACE to be translated into more than one ACE.

Use the following two guidelines to determine Layer 4 operation usage:

- Layer 4 operations are considered different if the operator or the operand differ. For example, in this ACL there are three different Layer 4 operations ("gt 10" and "gt 11" are considered two different Layer 4 operations):

```
... gt 10 permit
... lt 9 deny
... gt 11 deny
```

Note There is no limit to the use of "eq" operators as the "eq" operator does not use a logical operator unit (LOU) or a Layer 4 operation bit. See the "Determining Logical Operation Unit Usage" section on page 23-4 for a description of LOUs.

- Layer 4 operations are considered different if the same operator/operand couple applies once to a source port and once to a destination port. For example, in this ACL there are two different Layer 4 operations because one ACE applies to the source port and one applies to the destination port.

```
... Src gt 10 ...
... Dst gt 10
```

Determining Logical Operation Unit Usage

Logical operation units (LOUs) are registers that store operator-operand couples. All ACLs use LOUs. There can be up to 32 LOUs; each LOU can store two different operator-operand couples with the exception of the range operator. LOU usage per Layer 4 operation is as follows:

- gt uses 1/2 LOU
- lt uses 1/2 LOU
- neq uses 1/2 LOU
- range uses 1 LOU
- eq does not require a LOU

For example, this ACL would use a single LOU to store two different operator-operand couples:

```
... Src gt 10 ...
... Dst gt 10
```

A more detailed example follows:

```
ACL1
... (dst port) gt 10 permit
... (dst port) lt 9 deny
... (dst port) gt 11 deny
... (dst port) neq 6 permit
... (src port) neq 6 deny
... (dst port) gt 10 deny

ACL2
... (dst port) gt 20 deny
... (src port) lt 9 deny
... (src port) range 11 13 deny
... (dst port) neq 6 permit
```

The Layer 4 operations and LOU usage is as follows:

- ACL1 Layer 4 operations: 5
- ACL2 Layer 4 operations: 4
- LOUs: 4

An explanation of the LOU usage follows:

- LOU 1 stores “gt 10” and “lt 9”
- LOU 2 stores “gt 11” and “neq 6”
- LOU 3 stores “gt 20” (with space for one more)
- LOU 4 stores “range 11 13” (range needs the entire LOU)

Configuring the Cisco IOS Firewall Feature Set

Note Release 12.1(11b)E and later releases include firewall feature set images.

These sections describe configuring the Cisco IOS firewall feature set on the Cisco 7600 series routers:

- Cisco IOS Firewall Feature Set Support Overview, page 23-5

Cisco IOS Firewall Feature Set Support Overview

The firewall feature set images support these Cisco IOS firewall features:

- Context-based Access Control (CBAC)
- Port-to-Application Mapping (PAM)
- Authentication Proxy

These are the firewall feature set image names:

- c6sup22-jo3sv-mz
- c6sup22-po3sv-mz
- c6sup12-jo3sv-mz
- c6sup12-po3sv-mz

For more information about Cisco IOS firewall features, refer to the *Cisco IOS Security Configuration Guide, Release 12.1*, "Traffic Filtering and Firewalls" online publications:

- The "Cisco IOS Firewall Overview" chapter at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdfirwl.htm
- The "Configuring Context-Based Access Control" chapter at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdcbac.htm
- The "Configuring Authentication Proxy" chapter at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdauthp.htm
- Cisco IOS Security Command Reference publication at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_r/index.htm

The following features are supported with and without the use of a Cisco IOS firewall image:

- Standard access lists and static extended access lists
- Lock-and-key (dynamic access lists)
- IP session filtering (reflexive access lists)
- TCP intercept
- Security server support
- Network address translation
- Neighbor router authentication
- Event logging
- User authentication and authorization

Note Cisco 7600 series routers support the Intrusion Detection System Module (IDSM) (WS-X6381-IDS). Cisco 7600 series routers do not support the Cisco IOS firewall IDS feature, which is configured with the **ip audit** command.

Firewall Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring the Cisco IOS firewall features:

Restrictions

- On other platforms, if you enter the **ip inspect** command on a port, CBAC modifies ACLs on other ports to permit the inspected traffic to flow through the network device. On Cisco 7600 series routers, you must enter the **mls ip inspect** commands to permit traffic through any ACLs that would deny the traffic through other ports. See the "Configuring CBAC on Cisco 7600 Series Routers" section on page 23-6.
- With Supervisor Engine 2 and PFC2, reflexive ACLs and CBAC have conflicting flow mask requirements. When you configure CBAC on a switch with Supervisor Engine 2 and PFC2, reflexive ACLs are processed in software on the MSFC2.
- CBAC is incompatible with VACLs. You can configure both CBAC and VACLs on the switch but not in the same subnet (VLAN) or on the same interface.

Note The Intrusion Detection System Module (IDSM) uses VACLs to select traffic. To use the IDSM in a subnet where CBAC is configured, enter the **mls ip ids** *acl_name* interface command, where *acl_name* is configured to select traffic for the IDSM.

Guidelines

- To inspect Microsoft NetMeeting (2.0 or greater) traffic, turn on both **h323** and **tcp** inspection.
- To inspect web traffic, turn on **tcp** inspection. To avoid reduced performance, do not turn on **http** inspection to block Java.
- You can configure CBAC on physical ports configured as Layer 3 interfaces and on VLAN interfaces.
- QoS and CBAC do not interact or interfere with each other.

Configuring CBAC on Cisco 7600 Series Routers

You need to do additional CBAC configuration on the Cisco 7600 series routers. On a network device other than a Cisco 7600 series router, when ports are configured to deny traffic, CBAC permits traffic to flow bidirectionally through the port if it is configured with the **ip inspect** command. The same behavior applies to any other port that the traffic needs to go through, as shown in this example:

```
Router(config)# ip inspect name permit_ftp ftp
Router(config)# interface vlan 100
Router(config-if)# ip inspect permit_ftp in
Router(config-if)# ip access-group deny_ftp_a in
Router(config-if)# ip access-group deny_ftp_b out
```

```
Router(config-if)# exit
Router(config)# interface vlan 200
Router(config-if)# ip access-group deny_ftp_c in
Router(config-if)# ip access-group deny_ftp_d out
Router(config-if)# exit
Router(config)# interface vlan 300
Router(config-if)# ip access-group deny_ftp_e in
Router(config-if)# ip access-group deny_ftp_f out
Router(config-if)# end
```

If the FTP session enters on VLAN 100 and needs to leave on VLAN 200, CBAC permits the FTP traffic through ACLs deny_ftp_a, deny_ftp_b, deny_ftp_c, and deny_ftp_d. If another FTP session enters on VLAN 100 and needs to leave on VLAN 300, CBAC permits the FTP traffic through ACLs deny_ftp_a, deny_ftp_b, deny_ftp_e, and deny_ftp_f.

On a Cisco 7600 series router, when ports are configured to deny traffic, CBAC permits traffic to flow bidirectionally only through the port configured with the **ip inspect** command. You must configure other ports with the **mls ip inspect** command.

If the FTP session enters on VLAN 100 and needs to leave on VLAN 200, CBAC on a Cisco 7600 series router permits the FTP traffic only through ACLs deny_ftp_a and deny_ftp_b. To permit the traffic through ACLs deny_ftp_c and deny_ftp_d, you must enter the **mls ip inspect deny_ftp_c** and **mls ip inspect deny_ftp_d** commands, as shown in this example:

```
Router(config)# mls ip inspect deny_ftp_c
Router(config)# mls ip inspect deny_ftp_d
```

With the example configuration, FTP traffic cannot leave on VLAN 300 unless you enter the **mls ip inspect deny_ftp_e** and **mls ip inspect deny_ftp_f** commands. Enter the **show fm insp** [**detail**] command to verify the configuration.

The **show fm insp** [**detail**] command displays the list of ACLs and ports on which CBAC is configured and the status (**ACTIVE** or **INACTIVE**), as shown in this example:

```
Router# show fm insp
         interface:Vlan305(in) status :ACTIVE
         acl name:deny
           interfaces:
              Vlan305(out):status ACTIVE
```

On VLAN 305, inspection is active in the inbound direction and no ACL exists. ACL **deny** is applied on VLAN 305 in the outbound direction and inspection is active.

To display all of the flow information, use the **detail** keyword.

If a VACL is configured on the port before configuring CBAC, the status displayed is INACTIVE; otherwise, it is ACTIVE. If PFC resources are exhausted, the command displays the word “BRIDGE” followed by the number of currently active NetFlow requests that failed, which have been sent to the MSFC2 for processing.

Configuring MAC Address-Based Traffic Blocking

With 12.1(13)E and later releases, to block all traffic to or from a MAC address in a specified VLAN, perform this task:

Command	Purpose
`Router(config)# mac-address-table static mac_address vlan vlan_ID drop`	Blocks all traffic to or from the configured MAC address in the specified VLAN.
`Router(config)# no mac-address-table static mac_address vlan vlan_ID`	Clears MAC address-based blocking.

This example shows how to block all traffic to or from MAC address 0050.3e8d.6400 in VLAN 12:

```
Router# configure terminal
Router(config)# mac-address-table static 0050.3e8d.6400 vlan 12 drop
```

Configuring VLAN ACLs

Note Releases 12.1(11b)E or later supports VLAN ACLs (VACLs).

The following sections describe VACLs:

Understanding VACLs

These sections describe VACLs:

VACL Overview

VACLs can provide access control for all packets that are bridged within a VLAN or that are routed into or out of a VLAN or, with releases 12.1(13)E or later, a WAN interface for VACL capture. Unlike regular Cisco IOS standard or extended ACLs that are configured on router interfaces only and are applied on routed packets only, VACLs apply to all packets and can be applied to any VLAN or WAN interface. VACLS are processed in hardware. VACLs use Cisco IOS ACLs. VACLs ignore any Cisco IOS ACL fields that are not supported in hardware.

You can configure VACLs for IP, IPX, and MAC-Layer traffic. VACLs applied to WAN interfaces support only IP traffic for VACL capture.

When you configure a VACL and apply it to a VLAN, all packets entering the VLAN are checked against this VACL. If you apply a VACL to the VLAN and an ACL to a routed interface in the VLAN, a packet coming in to the VLAN is first checked against the VACL and, if permitted, is then checked against the input ACL before it is handled by the routed interface. When the packet is routed to another VLAN, it

is first checked against the output ACL applied to the routed interface and, if permitted, the VACL configured for the destination VLAN is applied. If a VACL is configured for a packet type and a packet of that type does not match the VACL, the default action is deny.

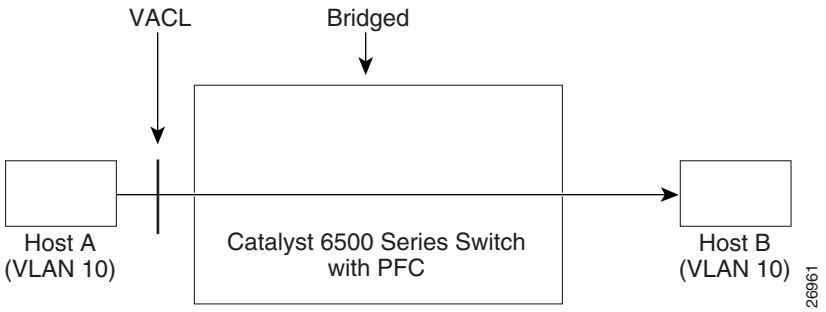
Note

- VACLs and CBAC cannot be configured on the same interface.
- TCP Intercepts and Reflexive ACLs take precedence over a VACL action if these are configured on the same interface.
- IGMP packets are not checked against VACLs.

Bridged Packets

Figure 23-1 shows a VACL applied on bridged packets.

Figure 23-1 Applying VACLs on Bridged Packets

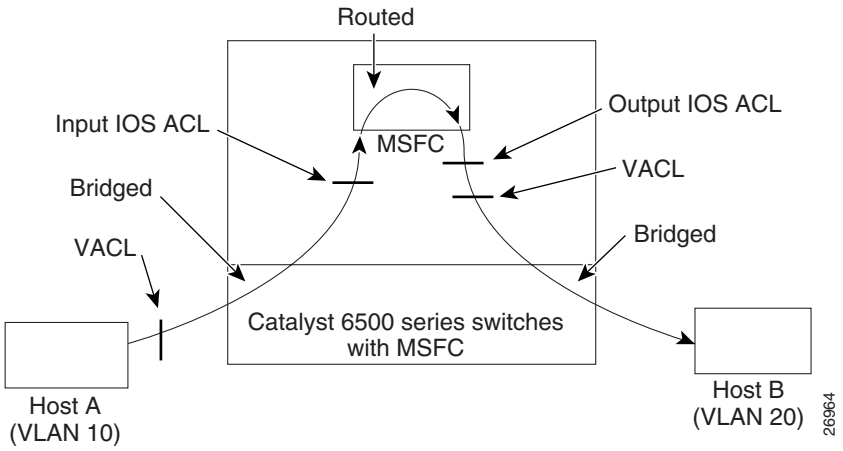


Routed Packets

Figure 23-2 shows how ACLs are applied on routed and Layer 3-switched packets. For routed or Layer 3-switched packets, the ACLs are applied in the following order:

1. VACL for input VLAN
2. Input Cisco IOS ACL
3. Output Cisco IOS ACL
4. VACL for output VLAN

Figure 23-2 Applying VACLs on Routed Packets

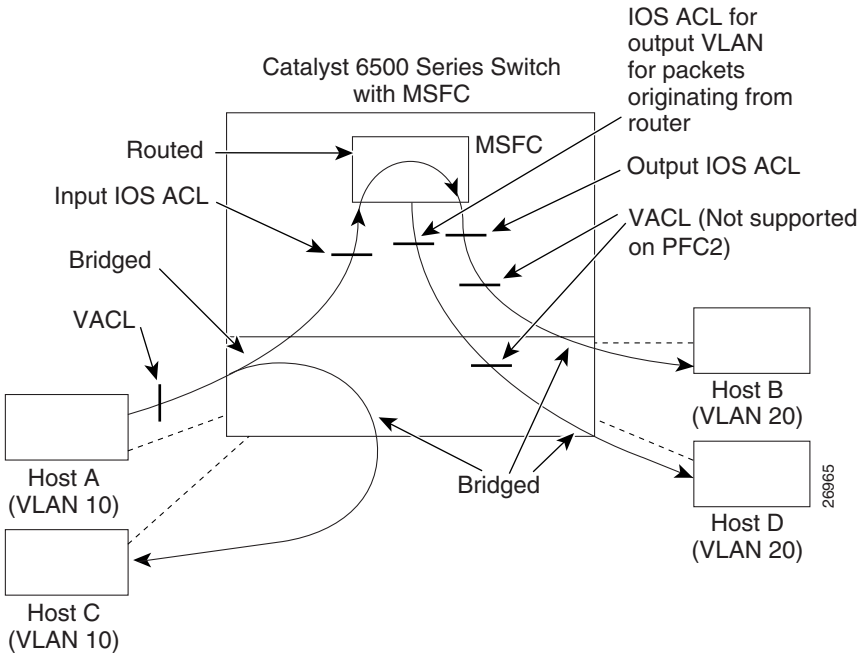


Multicast Packets

Figure 23-3 shows how ACLs are applied on packets that need multicast expansion. For packets that need multicast expansion, the ACLs are applied in the following order:

1. Packets that need multicast expansion:
   a. VACL for input VLAN
   b. Input Cisco IOS ACL
2. Packets after multicast expansion:
   a. Output Cisco IOS ACL
   b. VACL for output VLAN (not supported with PFC2)
3. Packets originating from router—VACL for output VLAN

Figure 23-3 Applying VACLs on Multicast Packets



Configuring VACLs

These sections describe configuring VACLs:

- VACL Configuration Overview, page 23-12
- Defining a VLAN Access Map, page 23-12
- Configuring a Match Clause in a VLAN Access Map Sequence, page 23-13
- Configuring an Action Clause in a VLAN Access Map Sequence, page 23-14
- Applying a VLAN Access Map, page 23-14
- Verifying VLAN Access Map Configuration, page 23-15

VACL Configuration Overview

VACLs use standard and extended Cisco IOS IP and IPX ACLs, and MAC-Layer named ACLs (see the "Configuring MAC-Layer Named Access Lists (Optional)" section on page 32-39) and VLAN access maps.

VLAN access maps can be applied to VLANs or, with releases 12.1(13)E or later, to WAN interfaces for VACL capture. VACLs attached to WAN interfaces support only standard and extended Cisco IOS IP ACLs for VACL capture.

Each VLAN access map can consist of one or more map sequences, each sequence with a match clause and an action clause. The match clause specifies IP, IPX, or MAC ACLs for traffic filtering and the action clause specifies the action to be taken when a match occurs. When a flow matches a permit ACL entry, the associated action is taken and the flow is not checked against the remaining sequences. When a flow matches a deny ACL entry, it will be checked against the next ACL in the same sequence or the next sequence. If a flow does not match any ACL entry and at least one ACL is configured for that packet type, the packet is denied.

To use access-control for both bridged and routed traffic, you can use VACLs alone or a combination of VACLs and ACLs. You can define ACLs on the VLAN interfaces to use access-control for both the input and output routed traffic. You can define a VACL to use access-control for the bridged traffic.

The following caveats apply to ACLs when used with VACLs:

- Packets that require logging on the outbound ACLs are not logged if they are denied by a VACL.
- VACLs are applied on packets before NAT translation. If the translated flow is not subject to access control, the flow might be subject to access control after the translation because of the VACL configuration.

The action clause in a VACL can be forward, drop, capture, or redirect. Traffic can also be logged. VACLs applied to WAN interfaces do not support the redirect or log actions.

Note VACLs have an implicit deny at the end of the map; a packet is denied if it does not match any ACL entry, and at least one ACL is configured for the packet type.

Note If an empty or undefined ACL is specified in a VACL, any packets will match the ACL and the associated action is taken.

Defining a VLAN Access Map

To define a VLAN access map, perform this task:

Command	Purpose
`Router(config)# vlan access-map map_name [0-65535]`	Defines the VLAN access map. Optionally, you can specify the VLAN access map sequence number.
`Router(config)# no vlan access-map map_name 0-65535`	Deletes a map sequence from the VLAN access map.
`Router(config)# no vlan access-map map_name`	Deletes the VLAN access map.

When defining a VLAN access map, note the following syntax information:

- To insert or modify an entry, specify the map sequence number.
- If you do not specify the map sequence number, a number is automatically assigned.
- You can specify only one match clause and one action clause per map sequence.
- Use the **no** keyword with a sequence number to remove a map sequence.
- Use the **no** keyword without a sequence number to remove the map.

See the "VLAN Access Map Configuration and Verification Examples" section on page 23-15.

Configuring a Match Clause in a VLAN Access Map Sequence

To configure a match clause in a VLAN access map sequence, perform this task:

Command	Purpose
`Router(config-access-map)# match {ip address {1-199 | 1300-2699 | acl_name} | ipx address {800-999 | acl_name}| mac address acl_name}`	Configures the match clause in a VLAN access map sequence.
`Router(config-access-map)# no match {ip address {1-199 | 1300-2699 | acl_name} | ipx address {800-999 | acl_name}| mac address acl_name}`	Deletes the match clause in a VLAN access map sequence.

When configuring a match clause in a VLAN access map sequence, note the following syntax information:

- You can select one or more ACLs.
- VACLs attached to WAN interfaces support only standard and extended Cisco IOS IP ACLs.
- Use the **no** keyword to remove a match clause or specified ACLs in the clause.
- For information about named MAC-Layer ACLs, refer to the "Configuring MAC-Layer Named Access Lists (Optional)" section on page 32-39.
- For information about Cisco IOS ACLs, refer to the *Cisco IOS Security Configuration Guide*, Release 12.1, "Traffic Filtering and Firewalls," "Access Control Lists: Overview and Guidelines," at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/index.htm

See the "VLAN Access Map Configuration and Verification Examples" section on page 23-15.

Configuring an Action Clause in a VLAN Access Map Sequence

To configure an action clause in a VLAN access map sequence, perform this task:

Command	Purpose
`Router(config-access-map)# action {drop [log]} | {forward [capture]} | {redirect {{ethernet | fastethernet | gigabitethernet | tengigabitethernet} slot/port} | {port-channel channel_id}}`	Configures the action clause in a VLAN access map sequence.
`Router(config-access-map)# no action {drop [log]} | {forward [capture]} | {redirect {{ethernet | fastethernet | gigabitethernet | tengigabitethernet} slot/port} | {port-channel channel_id}}`	Deletes the action clause in from the VLAN access map sequence.

When configuring an action clause in a VLAN access map sequence, note the following syntax information:

- You can set the action to drop, forward, forward capture, or redirect packets.
- VACLs applied to WAN interfaces support only the forward capture action. VACLs applied to WAN interfaces do not support the drop, forward, or redirect actions.
- Forwarded packets are still subject to any configured Cisco IOS security ACLs.
- The **capture** action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets. Only forwarded packets can be captured. For more information about the **capture** action, see the "Configuring a Capture Port" section on page 23-16.
- The **log** action is supported only on Supervisor Engine 2.
- VACLs applied to WAN interfaces do not support the **log** action.
- When the **log** action is specified, dropped packets are logged in software. Only dropped IP packets can be logged.
- The **redirect** action allows you to specify up to five interfaces, which can be physical interfaces or EtherChannels. You cannot specify packets to be redirected to an EtherChannel member or a VLAN interface.
- For systems with a Supervisor Engine 2, the redirect interface must be in the VLAN for which the VACL access map is configured. For systems with Supervisor Engine 1, the redirect interface must be in the redirected packet's source VLAN.
- Use the **no** keyword to remove an action clause or specified redirect interfaces.

See the "VLAN Access Map Configuration and Verification Examples" section on page 23-15.

Applying a VLAN Access Map

To apply a VLAN access map, perform this task:

Command	Purpose
`Router(config)# vlan filter map_name {vlan-list vlan_list | interface type[1] number[2]} CP_CmdPlain`	Applies the VLAN access map to the specified VLANs or WAN interfaces.

Command	Purpose
`Router(config)# no vlan filter map_name [vlan-list vlan_list | interface type[1] number[2]]`	Removes the VLAN access map from the specified VLANs or WAN interfaces.

1. *type* = **pos**, **atm**, or **serial**
2. number = *slot*/*port* or *slot*/*port_adapter*/*port*; can include a subinterface or channel group descriptor

When applying a VLAN access map, note the following syntax information:

- You can apply the VLAN access map to one or more VLANs or WAN interfaces.
- The *vlan_list* parameter can be a single VLAN ID or a comma-separated list of VLAN IDs or VLAN ID ranges (*vlan_ID*–*vlan_ID*).
- If you delete a WAN interface that has a VACL applied, the VACL configuration on the interface is also removed.
- You can apply only one VLAN access map to each VLAN or WAN interface.
- VACLs applied to VLANs are active only for VLANs with a Layer 3 VLAN interface configured. VACLs applied to VLANs without a Layer 3 VLAN interface are inactive. With releases 12.1(13)E and later, applying a VLAN access map to a VLAN without a Layer 3 VLAN interface creates an administratively down Layer 3 VLAN interface to support the VLAN access map. If creation of the Layer 3 VLAN interface fails, the VACL is inactive.
- You cannot apply a VACL to a secondary private VLAN. VACLs applied to primary private VLANs also apply to secondary private VLANs.
- Use the **no** keyword to clear VLAN access maps from VLANs or WAN interfaces.

See the "VLAN Access Map Configuration and Verification Examples" section on page 23-15.

Verifying VLAN Access Map Configuration

To verify VLAN access map configuration, perform this task:

Command	Purpose
`Router# show vlan access-map [map_name]`	Verifies VLAN access map configuration by displaying the content of a VLAN access map.
`Router# show vlan filter [access-map map_name | vlan vlan_id | interface type[1] number[2]]`	Verifies VLAN access map configuration by displaying the mappings between VACLs and VLANs.

1. *type* = **pos**, **atm**, or **serial**
2. number = *slot*/*port* or *slot*/*port_adapter*/*port*; can include a subinterface or channel group descriptor

VLAN Access Map Configuration and Verification Examples

Assume IP-named ACL **net_10** and **any_host** are defined as follows:

```
Router# show ip access-lists net_10
Extended IP access list net_10
    permit ip 10.0.0.0 0.255.255.255 any

Router# show ip access-lists any_host
Standard IP access list any_host
    permit any
```

This example shows how to define and apply a VLAN access map to forward IP packets. In this example, IP traffic matching net_10 is forwarded and all other IP packets are dropped due to the default drop action. The map is applied to VLAN 12 to 16.

```
Router(config)# vlan access-map thor 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action forward
Router(config-access-map)# exit
Router(config)# vlan filter thor vlan-list 12-16
```

This example shows how to define and apply a VLAN access map to drop and log IP packets. In this example, IP traffic matching net_10 is dropped and logged and all other IP packets are forwarded:

```
Router(config)# vlan access-map ganymede 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action drop log
Router(config-access-map)# exit
Router(config)# vlan access-map ganymede 20
Router(config-access-map)# match ip address any_host
Router(config-access-map)# action forward
Router(config-access-map)# exit
Router(config)# vlan filter ganymede vlan-list 7-9
```

This example shows how to define and apply a VLAN access map to forward and capture IP packets. In this example, IP traffic matching net_10 is forwarded and captured and all other IP packets are dropped:

```
Router(config)# vlan access-map mordred 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action forward capture
Router(config-access-map)# exit
Router(config)# vlan filter mordred vlan-list 2, 4-6
```

Configuring a Capture Port

A port configured to capture VACL-filtered traffic is called a capture port.

Note To apply IEEE 802.1Q or ISL tags to the captured traffic, configure the capture port to trunk unconditionally (see the "Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk" section on page 7-8 and the "Configuring the Layer 2 Trunk Not to Use DTP" section on page 7-9).

To configure a capture port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{type[1] slot/port}`	Specifies the interface to configure.
Step 2	`Router(config-if)# switchport capture allowed vlan {add | all | except | remove} vlan_list`	(Optional) With Release 12.1(13)E and later releases, filters the captured traffic on a per-destination-VLAN basis. The default is **all**.
	`Router(config-if)# no switchport capture allowed vlan`	Clears the configured destination VLAN list and returns to the default value (**all**).
Step 3	`Router(config-if)# switchport capture`	Configures the port to capture VACL-filtered traffic.
	`Router(config-if)# no switchport capture`	Disables the capture function on the interface.

1. *type* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring a capture port, note the following syntax information:

- With Release 12.1(13)E and later releases, you can configure any port as a capture port. With earlier releases, only the Gigabit Ethernet monitor port on the IDS module can be configured as a capture port.
- When configuring a capture port with Release 12.1(13)E and later releases, note the following syntax information:
  - The *vlan_list* parameter can be a single VLAN ID or a comma-separated list of VLAN IDs or VLAN ID ranges (*vlan_ID–vlan_ID*).
  - To encapsulate captured traffic, configure the capture port with the **switchport trunk encapsulation** command (see the "Configuring a Layer 2 Switching Port as a Trunk" section on page 7-8) before you enter the **switchport capture** command.
  - To not encapsulate captured traffic, configure the capture port with the **switchport mode access** command (see the "Configuring a LAN Interface as a Layer 2 Access Port" section on page 7-14) before you enter the **switchport capture** command.
  - The capture port supports only egress traffic. No traffic can enter the router through a capture port.

This example shows how to configure a Fast Ethernet interface 5/1 as a capture port:

```
Router(config)# interface gigabitEthernet 5/1
Router(config-if)# switchport capture
Router(config-if)# end
```

This example shows how to display VLAN access map information:

```
Router# show vlan access-map mordred
Vlan access-map "mordred"  10
        match: ip address net_10
        action: forward capture
Router#
```

This example shows how to display mappings between VACLs and VLANs. For each VACL map, there is information about the VLANs that the map is configured on and the VLANs that the map is active on. A VACL is not active if the VLAN does not have an interface.

```
Router# show vlan filter
VLAN Map mordred:
        Configured on VLANs:  2,4-6
            Active on VLANs:  2,4-6
Router#
```

Configuring VACL Logging

When you configure VACL logging, IP packets that are denied generate log messages in these situations:

- When the first matching packet is received
- For any matching packets received during the last 5-minute interval
- If the threshold is reached before the 5-minute interval

Log messages are generated on a per-flow basis. A flow is defined as packets with the same IP addresses and Layer 4 (UDP or TCP) port numbers. When a log message is generated, the timer and packet count is reset.

These restrictions apply to VACL logging:

- Supported only with Supervisor Engine 2.
- Because of the rate-limiting function for redirected packets, VACL logging counters may not be accurate.
- Only denied IP packets are logged.

To configure VACL logging, use the **action drop log** command action in VLAN access map submode (see the "Configuring VACLs" section on page 23-11 for configuration information) and perform this task in global configuration mode to specify the global VACL logging parameters:

	Command	Purpose
Step 1	`Router(config)# vlan access-log maxflow` *max_number*	Sets the log table size. The content of the log table can be deleted by setting the maxflow number to 0. The default is 500 with a valid range of 0 to 2048. When the log table is full, logged packets from new flows are dropped by the software.
Step 2	`Router(config)# vlan access-log ratelimit` *pps*	Sets the maximum redirect VACL logging packet rate. The default packet rate is 2000 packets per second with a valid range of 0 to 5000. Packets exceeding the limit are dropped by the hardware.
Step 3	`Router(config)# vlan access-log threshold` *pkt_count*	Sets the logging threshold. A logging message is generated if the threshold for a flow is reached before the 5-minute interval. By default, no threshold is set.
Step 4	`Router(config)# exit`	Exits VLAN access map configuration mode.
Step 5	`Router# show vlan access-log config`	(Optional) Displays the configured VACL logging properties.
Step 6	`Router# show vlan access-log flow` *protocol* {{*src_addr src_mask*} | **any** | {**host** {*hostname* | *host_ip*}}} {{*dst_addr dst_mask*} | **any** | {**host** {*hostname* | *host_ip*}}} [**vlan** *vlan_id*]	(Optional) Displays the content of the VACL log table.
Step 7	`Router# show vlan access-log statistics`	(Optional) Displays packet and message counts and other statistics.

This example shows how to configure global VACL logging in hardware:

```
Router(config)# vlan access-log maxflow 800
Router(config)# vlan access-log ratelimit 2200
Router(config)# vlan access-log threshold 4000
```

Configuring TCP Intercept

With Supervisor Engine 2 and PFC2, TCP intercept flows are processed in hardware.

With Supervisor Engine 1 and PFC, TCP intercept flows are processed in software.

For configuration procedures, refer to the *Cisco IOS Security Configuration Guide*, Release 12.1, "Traffic Filtering and Firewalls," "Configuring TCP Intercept," at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scddenl.htm

Configuring Unicast Reverse Path Forwarding

These sections describe configuring Cisco IOS Unicast Reverse Path Forwarding (Unicast RPF):

Understanding Unicast RPF Support

The PFC2 supports Unicast RPF with hardware processing for packets that have a single return path. The MSFC2 processes traffic in software that has multiple return paths (for example, load sharing).

With a PFC2, if you configure Unicast RPF to filter with an ACL, the PFC2 determines whether or not traffic matches the ACL. The PFC2 sends the traffic denied by the RPF ACL to the MSFC2 for the Unicast RPF check.

Note

- Because the packets in a denial-of-service attack typically match the deny ACE and are sent to the MSFC2 for the unicast RPF check, they can overload the MSFC2.
- The PFC2 provides hardware support for traffic that does not match the Unicast RPF ACL, but that does match an input security ACL.

With Supervisor Engine 1 and PFC, the MSFC or MSFC 2 supports Unicast RPF in software.

Configuring Unicast RPF

For configuration procedures, refer to the *Cisco IOS Security Configuration Guide*, Release 12.1, "Other Security Features," "Configuring Unicast Reverse Path Forwarding" at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt5/scdrpf.htm

Enabling Self-Pinging

With Unicast RPF enabled, the router cannot ping itself. To enable self-pinging, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{vlan vlan_ID} | {type[1] slot/port} | {port-channel number}}`	Selects the interface to configure.
Step 2	`Router(config-if)# ip verify unicast source reachable-via any allow-self-ping`	Enables the router to ping itself or a secondary address.
	`Router(config-if)# no ip verify unicast source reachable-via any allow-self-ping`	Disables self-pinging.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode.

1. *type* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable self-pinging:

```
Router(config)# interface gigabitethernet 4/1
Router(config-if)# ip verify unicast source reachable-via any allow-self-ping
Router(config-if)# end
```

Configuring the Unicast RPF Checking Mode

There are two Unicast RPF checking modes:

- Strict checking mode, which verifies that the source IP address exists in the FIB table and verifies that the source IP address is reachable through the input port.
- Exist-only checking mode, which only verifies that the source IP address exists in the FIB table.

Note The most recently configured mode is automatically applied to all ports configured for Unicast RPF checking.

To configure Unicast RPF checking mode, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{vlan vlan_ID} | {type[1] slot/port} | {port-channel number}}`	Selects an interface to configure. **Note** Based on the input port, Unicast RPF verifies the best return path before forwarding the packet on to the next destination.
Step 2	`Router(config-if)# ip verify unicast source reachable-via {rx | any} [allow-default] [list]`	Configures the Unicast RPF checking mode.
	`Router(config-if)# no ip verify unicast`	Reverts to the default Unicast RPF checking mode.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode.

1. *type* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring the Unicast RPF checking mode, note the following syntax information:

- Use the **rx** keyword to enable strict checking mode.
- Use the **any** keyword to enable exist-only checking mode.
- Use the **allow-default** keyword to allow use of the default route for RPF verification.
- Use the *list* option to identify an access list.
  - If the access list denies network access, spoofed packets are dropped at the port.
  - If the access list permits network access, spoofed packets are forwarded to the destination address. Forwarded packets are counted in the interface statistics.
  - If the access list includes the logging action, information about the spoofed packets is sent to the log server.

Note When you enter the **ip verify unicast source reachable-via** command, the Unicast RPF checking mode changes on all ports in the router.

This example shows how to enable Unicast RPF exist-only checking mode on Gigabit Ethernet port 4/1:

```
Router(config)# interface gigabitethernet 4/1
Router(config-if)# ip verify unicast source reachable-via any
Router(config-if)# end
Router#
```

This example shows how to enable Unicast RPF strict checking mode on Gigabit Ethernet port 4/2:

```
Router(config)# interface gigabitethernet 4/2
Router(config-if)# ip verify unicast source reachable-via rx
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show running-config interface gigabitethernet 4/2
Building configuration...
Current configuration : 114 bytes
!
interface GigabitEthernet4/2
ip address 42.0.0.1 255.0.0.0
ip verify unicast reverse-path
no cdp enable
end
Router# show running-config interface gigabitethernet 4/1
Building configuration...
Current configuration : 114 bytes
!
interface GigabitEthernet4/1
ip address 41.0.0.1 255.0.0.0
→ ip verify unicast reverse-path (RPF mode on g4/1 also changed to strict-check RPF mode)
no cdp enable
end
Router#
```

Configuring Unicast Flood Protection

The unicast flood protection feature protects the system from disruptions caused by unicast flooding. The Cisco 7600 series routers use forwarding tables to direct traffic to specific ports based on the VLAN number and the destination MAC address of the frame. When there is no entry corresponding to the frame's destination MAC address in the incoming VLAN, the frame is sent to all forwarding ports within the respective VLAN, which causes flooding. Limited flooding is part of the normal switching process, but continuous flooding can cause adverse performance effects on the network.

When you enable the unicast flood protection feature, the system sends an alert when the rate limit has been exceeded, filters the traffic, or shuts down the port generating the floods when it detects unknown unicast floods exceeding a threshold.

To configure unicast flood protection, perform this task:

	Command	Purpose
Step 1	`Router(config)# [no] mac-address-table unicast-flood {limit kfps} {vlan vlan} {filter timeout | alert | shutdown}`	Enables unicast flood protection globally.
Step 2	`Router# show mac-address-table unicast-flood`	Displays unicast flood protection information.

When configuring unicast flood protection, note the following syntax information:

- Use the **limit** keyword to specify the unicast floods on a per source MAC address and per VLAN basis; valid values are from 1 to 4000 floods per second (fps).
- Use the **filter** keyword to specify how long to filter unicast flood traffic; valid values are from 1 to 34560 minutes.
- Use the **alert** keyword to configure the system to send an alert message when frames of unicast floods exceed the flood rate limit.
- Use the **shutdown** keyword to configure the system to shut down the ingress port generating the floods when frames of unicast floods exceed the flood rate limit.

This example shows how to configure the system to filter unicast flood traffic for 5 minutes and set the flood rate limit to 3000 fps:

```
Router(config)# mac-address-table unicast-flood limit 3 vlan 100 filter 5
Router # show mac-address-table unicast-flood
Unicast Flood Protection status: enabled

Configuration:
 vlan      Kfps         action       timeout
------+----------+----------------+----------
  100          3             filter         5

Mac filters:
 No.   vlan   souce mac addr.           installed on           time left (mm:ss)
-----+------+-----------------+------------------------------+------------------

Router(config)#
```

Configuring MAC Move Notification

When you configure MAC move notification, a message is generated when a MAC address moves from one port to another.

Note The MAC address move notification feature does not generate a notification when a new MAC address is added to the CAM or when a MAC address is removed from the CAM.

To configure MAC move notification, perform this task:

	Command	Purpose
Step 1	`Router(config)# [no] mac-address-table notification mac-move`	Enables MAC move notification globally.
Step 2	`Router# show mac-address-table notification mac-move`	Displays MAC move notification information.

This example shows how to enable the MAC move notification feature:

```
Router(config)# mac-address-table notification mac-move
Router# show mac-address-table notification mac-move
MAC Move Notification: enabled
Router#
```